IMPLEMENTATION OF VARIOUS CONTROL STRUCTURES USING PL/SQL

AIM:

14.Implementation of various control structures like IF-THEN, IF-THEN-ELSE, IF-THENELSIF,

CASE, WHILE using PL/SQL.

* To perform addition, subtraction, multiplication and division on two numbers.



2.To print the first ‘n’ prime numbers.



3.Display the Fibonacci series up to ‘n’ terms



factorial of a number

--------------------------

set serveroutput on;

declare

num number := 6;

fact number := 1;

temp number;

begin

temp :=num;

while( temp>0 )

loop

fact := fact\*temp;

temp := temp-1;

end loop;

dbms\_output.put\_line('factorial of '|| num || ' is ' || fact);

end;

/

factorial of any number

------------------------

declare

fac number :=1;

n number := &1;

begin

while n > 0 loop

fac:=n\*fac;

n:=n-1;

end loop;

dbms\_output.put\_line(fac);

end;

/

additin of a numbers

----------------------------

declare

-- declare variable x, y

-- and z of datatype number

x number(5);

y number(5);

z number(7);

begin

-- Here we Assigning 10 into x

x:=10;

-- Assigning 20 into x

y:=20;

-- Assigning sum of x and y into z

z:=x+y;

-- Print the Result

dbms\_output.put\_line('Sum is '||z);

end;

/

sum=30

additin of any two numbers

-----------------------------

Declare

Var1 integer;

Var2 integer;

Var3 integer;

Begin

Var1:=&var1;

Var2:=&var2;

Var3:=var1+var2;

Dbms\_output.put\_line(var3);

End;

/



Introduction

In this article, we will learn how to create a table, insert records, select statements, update a statement, truncate a table and drop table statements in an oracle database.

1. Create Table

Create Table statement used to create the new table in the oracle database, in the following code, we will create an Employee table in the oracle database. in the following script, we will create an Employee table with having columns EmployeeNo, EmployeeName, EmployeeSal, EmployeeCity, EmployeeDob.

CREATE TABLE Employee

(

EmployeeNo char(4),

EmployeeName varchar2(30),

EmployeeSal number(10,2),

EmployeeCity varchar2(30),

EmployeeDob date

);

2. Insert Record

INSERT statement used to insert values into the table, we can insert optional column by specifying the column name. using the following technique we will insert an optional column into a table. In below script we will insert EmployeeNo, EmployeeName, EmployeeSal, EmployeeCity, EmployeeDob data into the Employee table.

INSERT INTO Employee(EmployeeNo,EmployeeName,EmployeeSal,EmployeeCity,EmployeeDob)

values('1', 'Arvind', 5000, 'Mumbai','23-DEC-1992');

INSERT INTO Employee

values('2', 'Santosh', 5000, 'Delhi','23-DEC-1994');

3. Select Statement

The select statement is used to select the record from the table, either we can use \* (all the columns) or specify the column name. In the following example, we will select all the records into the table

select \* from Employee

or

(In the below example we will specify the number of columns, the best practice always specify the column name )

select EmployeeNo,EmployeeName,EmployeeSal,EmployeeCity,EmployeeDob from Employee

4. Update Statement

with the help of the update statement, we can update the records into a table. In the below example we will update the employee name KASHISH having employeeNo 1.

UPDATE Employee

SET EmployeeName='KASHISH'

WHERE EmployeeNo=1

Delete Statement

with the help of the delete statement, we can delete the records from the table. in the following script, we will delete the records from the Employee table having EmployeeNo 1.

DELETE

FROM

EMPLOYEE

WHERE employeeNo=1

6. Truncate Statement

The following syntax for the TRUNCATE TABLE statement in Oracle/PLSQL . In the following example, we will truncate the employee table

TRUNCATE TABLE EMPLOYEE;

7. Drop Table Statement

Oracle DROP TABLE statement is used to remove or delete a table from the Oracle database. The following statement will drop the table named Employee.

DROP TABLE EMPLOYEE;





**mangalam**

**pl/sql**

set serveroutput on

begin

dbms\_output.Put\_line('hai');

end;

/

**hai**

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

declare

var1 varchar(10);

num1 int(30);

begin

var1:='hello';

num1:=100;

dbms\_output.Put\_line('var1:'||var1);

dbms\_output.Put\_line('Num1:'||num1);

end;

/

var1:hello

Num1:100

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

set serveroutput on

CREATE OR REPLACE PROCEDURE greetings

AS

BEGIN

dbms\_output.put\_line('Hello World!');

END;

/

Procedure created.

SQL> execute greetings

Hello World!

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

create table Employees(eid int,fname varchar(30),salary int);

insert into Employees values(100,'ram',30000);

1 row created.

insert into Employees values(101,'sam',4000);

1 row created.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

declare

name varchar(10);

sal int(20);

begin

select fname,salary into name,sal from Employees where eid=100;

dbms\_output.Put\_line('Name:'||name);

dbms\_output.Put\_line('Salary:'||sal);

end;

/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

declare

name Employees.fname%TYPE;

sal Employees.salary%TYPE;

lname name%TYPE;

begin

select fname,salary into name,sal from Employees where eid=100;

dbms\_output.Put\_line('Name:'||name);

dbms\_output.Put\_line('Salary:'||sal);

end;

/

Name:ram

Salary:30000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

FACTORIAL OF A NUMBER

set serveroutput on;

declare

num number := 6;

fact number := 1;

temp number;

begin

temp :=num;

while( temp>0 )

loop

fact := fact\*temp;

temp := temp-1;

end loop;

dbms\_output.put\_line('factorial of '|| num || ' is ' || fact);

end;

/

factorial of 6 is 720

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

FACTORIAL OF ANY NUMBER

declare

n number;

fac number:=1;

i number;

begin

n:=&n;

for i in 1..n

loop

fac:=fac\*i;

end loop;

dbms\_output.put\_line('factorial='||fac);

end;

/

Enter value for n: 5

old 7: n:=&n;

new 7: n:=5;

factorial=120

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ADDITION OF 2 NUMBERS

declare

x number(5);

y number(5);

z number(7);

begin

x:=10;

y:=20;

z:=x+y;

dbms\_output.put\_line('Sum is '||z);

end;

/

Sum is 30

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ADDITION OF ANY TWO NUMBERS

Declare

Var1 integer;

Var2 integer;

Var3 integer;

Begin

Var1:=&var1;

Var2:=&var2;

Var3:=var1+var2;

Dbms\_output.put\_line(var3);

End;

/

Enter value for var1: 5

old 6: Var1:=&var1;

new 6: Var1:=5;

Enter value for var2: 6

old 7: Var2:=&var2;

new 7: Var2:=6;

11

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

begin

dbms\_output.put\_line('This is my first PL/SQL Program');

dbms\_output.put\_line('PL/SQL is a procedural Language');

dbms\_output.put\_line('I like PL/SQL');

end;

/

This is my first PL/SQL Program

PL/SQL is a procedural Language

I like PL/SQL

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

declare

num1 number;

num2 number;

num3 number;

begin

num1:=20;

num2:=15;

num3:=num1+num2;

dbms\_output.put\_line('The sum of '||num1||' and '||num2||' is '||num3);

num3:=num1-num2;

dbms\_output.put\_line('The subtraction of '||num1||' and '||num2||' is'||num3);

num3:=num1\*num2;

dbms\_output.put\_line('The multiplication of '||num1||' and '||num2||'is '||num3);

num3:=num1/num2;

dbms\_output.put\_line('The division of '||num1||' and '||num2||' is '||num3);

end;

/

The sum of 20 and 15 is 35

The subtraction of 20 and 15 is5

The multiplication of 20 and 15is 300

The division of 20 and 15 is 1.33333333333333333333333333333333333333

PL/SQL procedure successfully completed.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Write a PL/SQL program to display addition, subtraction, multiplication and division of two numbers and numbers will be entered by user.

declare

num1 number;

num2 number;

num3 number;

begin

num1:=&num1;

num2:=&num2;

num3:=num1+num2;

dbms\_output.put\_line('The sum of '||num1||' and '||num2||' is '||num3);

num3:=num1-num2;

dbms\_output.put\_line('The subtraction of '||num1||' and '||num2||' is '||num3);

num3:=num1\*num2;

dbms\_output.put\_line('The multiplication of '||num1||' and '||num2||'is '||num3);

num3:=num1/num2;

dbms\_output.put\_line('The division of '||num1||' and '||num2||' is '||num3);

end;

/

Enter value for num1: 4

old 6: num1:=&num1;

new 6: num1:=4;

Enter value for num2: 5

old 7: num2:=&num2;

new 7: num2:=5;

The sum of 4 and 5 is 9

The subtraction of 4 and 5 is -1

The multiplication of 4 and 5is 20

The division of 4 and 5 is .8

PL/SQL procedure successfully completed.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Write a PL/SQL program to check whether entered number is even or odd using if-else statement.

declare

x number;

begin

x:=&number;

if mod(x,2)=0 then

dbms\_output.put\_line('The number '||x||'is even');

else

dbms\_output.put\_line('The number '||x||'is odd');

end if;

end;

/

OUTPUT -

Enter value for number: 5

old 4: x:=&number;

new 4: x:=5;

The number 5 is odd

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Write a PL/SQL program to find largest number of three numbers using if statement.

declare

x number;

y number;

z number;

begin

x:=&number1;

y:=&number2;

z:=&number3;

if x>y and x>z then

dbms\_output.put\_line('x is largest');

end if;

if y>x and y>z then

dbms\_output.put\_line('y is largest');

end if;

if z>x and z>y then

dbms\_output.put\_line('y is largest');

end if;

end;

/

Enter value for number1: 4

old 6: x:=&number1;

new 6: x:=4;

Enter value for number2: 5

old 7: y:=&number2;

new 7: y:=5;

Enter value for number3: 1

old 8: z:=&number3;

new 8: z:=1;

y is largest

PL/SQL procedure successfully completed.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Write a PL/SQL program to find largest number of three numbers using if-elsif statement.

declare

x number;

y number;

z number;

begin

x:=&number1;

y:=&number2;

z:=&number3;

if x>y and x>z then

dbms\_output.put\_line('x is largest');

elsif y>x and y>z then

dbms\_output.put\_line('y is largest');

else

dbms\_output.put\_line('z is largest');

end if;

end;

/

Enter value for number1: 7

old 7: x:=&number1;

new 7: x:=7;

Enter value for number2: 2

old 8: y:=&number2;

new 8: y:=2;

Enter value for number3: 3

old 9: z:=&number3;

new 9: z:=3;

x is largest

PL/SQL procedure successfully completed.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**PROCEDURE**

CREATE [OR REPLACE] PROCEDURE procedure\_name

[(parameter\_name [IN | OUT | IN OUT] type [, ...])]

{IS | AS}

BEGIN

< procedure\_body >

END procedure\_name;

set serveroutput on

CREATE OR REPLACE PROCEDURE greetings

AS

BEGIN

dbms\_output.put\_line('Hello World!');

END;

/

Procedure created.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SQL> execute greetings

Hello World!

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CREATE OR REPLACE PROCEDURE test1 IS

2 BEGIN

3 dbms\_output.put\_line('test sucess');

4 END;

5 /

Procedure created.

EXECUTE test1;

test sucess

PL/SQL procedure successfully completed.

CREATE OR REPLACE PROCEDURE addemp

(depid IN dept.eid%TYPE,

depname IN dept.ename%TYPE,

depsal IN dept.psal%TYPE,

deploc IN dept.ploc%TYPE)

IS

BEGIN

INSERT INTO dept Values(depid,depname,depsal,deploc);

dbms\_output.Put\_line('dpmt added');

end;

/

procedure created

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

execute addemp(2,'MARY',3000,'klm');

dpmt added

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

BEGIN

greetings;

END;

/

Hello World!

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DROP PROCEDURE procedure-name;

DROP PROCEDURE greetings;

Procedure dropped.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DECLARE

a number;

b number;

c number;

PROCEDURE findMin(x IN number, y IN number, z OUT number) IS

BEGIN

IF x < y THEN

z:= x;

ELSE

z:= y;

END IF;

END;

BEGIN

a:= 31;

b:= 56;

findMin(a, b, c);

dbms\_output.put\_line(' Minimum of (31,56: ' || c);

END;

/

Minimum of (31,56): 31

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DECLARE

a number;

PROCEDURE squareNum(x IN OUT number) IS

BEGIN

x := x \* x;

END;

BEGIN

a:= 21;

squareNum(a);

dbms\_output.put\_line(' Square of (21): ' || a);

END;

/

Square of (21): 441

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DELIMITER //

CREATE PROCEDURE sumcalc()

BEGIN

DECLARE num1 INT;

DECLARE num2 INT;

DECLARE num3 INT;

DECLARE total INT;

SET num1 = 10;

SET num2 = 20;

SET num3 = 30;

SET total = num1 + num2 + num3;

SELECT total AS sum\_result;

END;

/

DELIMITER ;

**FUNCTIONS**

CREATE [OR REPLACE] FUNCTION function\_name

[(parameter\_name [IN | OUT | IN OUT] type [, ...])]

RETURN return\_datatype

{IS | AS}

BEGIN

< function\_body >

END [function\_name];

Where,

function-name specifies the name of the function.

[OR REPLACE] option allows the modification of an existing function.

The optional parameter list contains name, mode and types of the parameters. IN represents the value that will be passed from outside and OUT represents the parameter that will be used to return a value outside of the procedure.

The function must contain a return statement.

The RETURN clause specifies the data type you are going to return from the function.

function-body contains the executable part.

The AS keyword is used instead of the IS keyword for creating a standalone function.

create function sqarea1(len int)return int is

2 area int(5,3);

3 begin

4 area:=(len\*len);

5 return area;

6 end;

7 /

Function created

SQL> begin

2 dbms\_output.Put\_line('area='||sqarea1(4));

3 end;

4 /

area=16

PL/SQL procedure successfully completed